

Matlab et Arduino

Matlab et Arduino

Introduction

Le choix académique actuel en matière de logiciel de simulation multi-physique se porte actuellement sur Matlab.

Il convient alors de s'ouvrir aux possibilités actuelles que cette solution logicielle permet à ce jour

en matière de communication avec l'extérieur d'autant que la situation évolue de jour en jour...

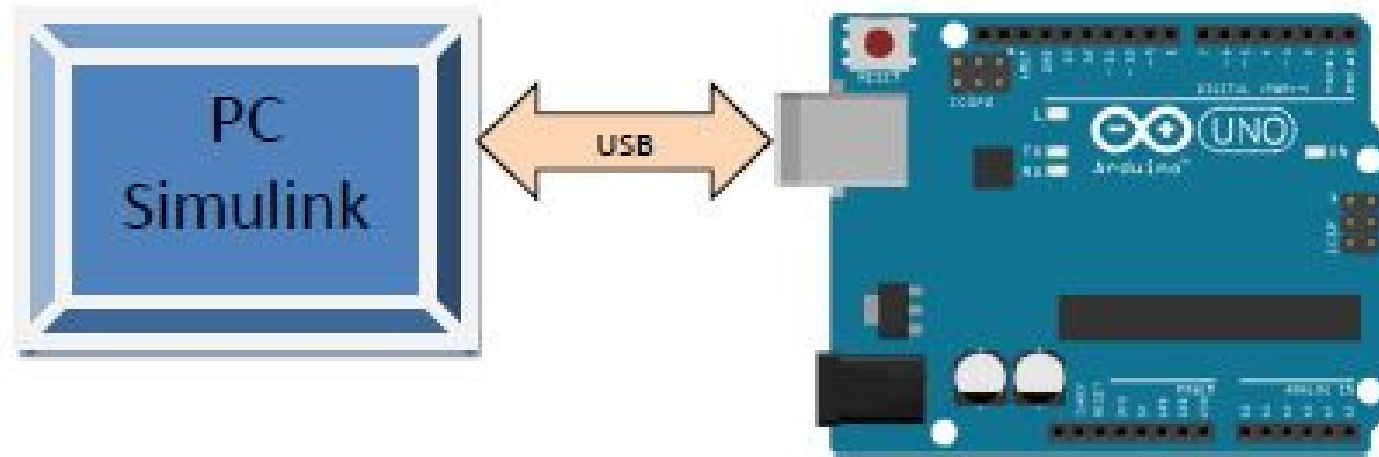
Il existe trois possibilités d'interfaçage d'arduino avec Matlab/Simulink.

- 1. Programmation de la carte Arduino Uno comme une carte d'interface.**
- 2. Utilisation du package ArduinoIO.**
- 3. Utilisation du package Arduino Target**

Matlab et Arduino

Programmation de la carte Arduino Uno comme une carte d'interface.

L'idée est d'utiliser les fonctions disponibles dans le langage Arduino pour envoyer et d'acquérir des données binaires via le port série (USB). Ainsi, sous simulink nous pourrons développer des programmes permettant de visualiser, stocker et/ou traiter ces données.



Matlab et Arduino

Programmation de la carte Arduino Uno comme une carte d'interface.

Configuration de l'arduino

Il s'agit tout simplement de configurer, comme nous l'avons vu précédemment, l'arduino pour une transmission série.

On doit déclarer l'objet Serial. Ensuite on utilisera des fonctions d'envoi et de lecture comme par exemple:

- `available()`: Permet d'obtenir le nombre de bit (caractères) disponibles pour lire du port série. Ces données sont stockées dans le buffer qui peut sauvegarder 64 bit.
- `read()`: Permet la lecture des bits entrants sur le port série(acquisition des données).
- `write()`: Permet l'écriture des bits sur le port série.(envoi des données)

Matlab et Arduino

Programmation de la carte Arduino Uno comme une carte d'interface.

Le programme suivant assure l'échange de données via le port série (USB) :

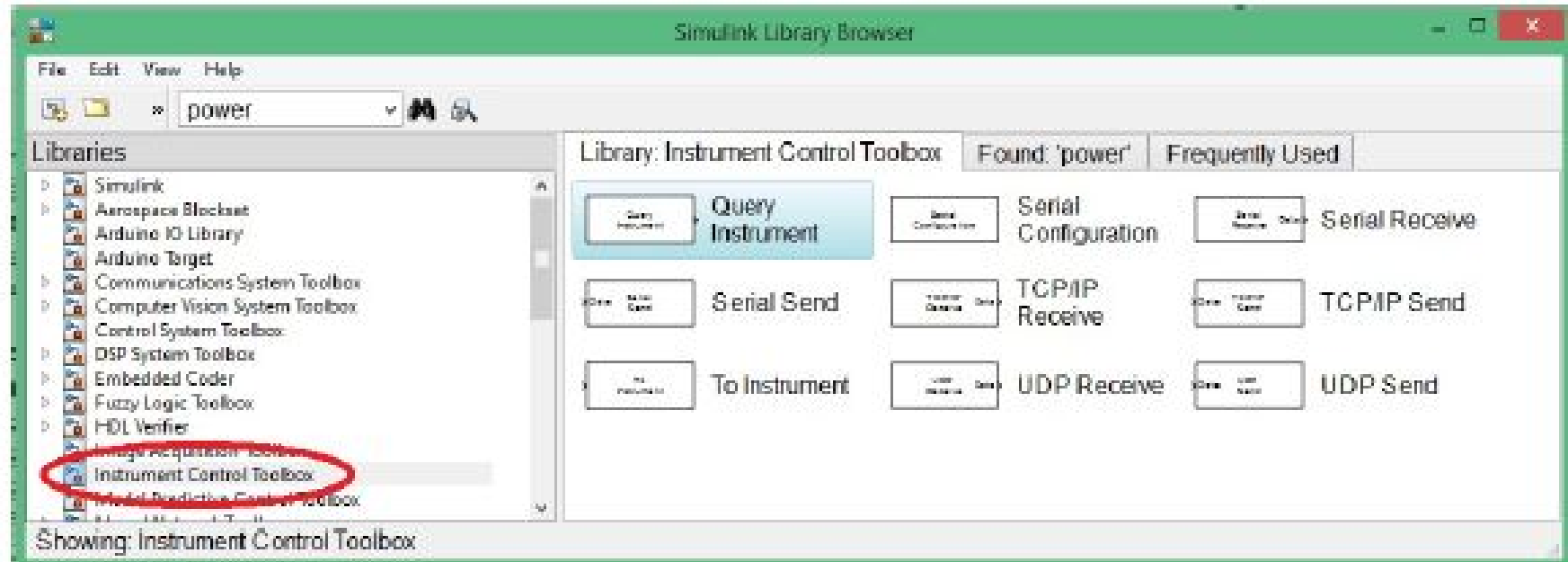
```
int entree ; // entree CAN
int sortie ; // sortie
void setup ()
{
Serial.begin (9600) ; // ouvre le port serie , fixe le debit a 9600 bauds
pinMode (6 ,OUTPUT) ; // Configuration de la pin 6 comme sortie
}
void loop ()
{
entree=analogRead (A0) ; // lecture du CAN ( valeur entre 0 et 1024)
Serial.write ( entree ) ; // Envoie de la donnee sur le port USB
if ( Serial.available ()) // si des donnees entrantes sont presentes
{
sortie=Serial.read () ; // lecture des donnees arrivees
analogWrite (6 , sortie ) ; // Transfert de ces donnees sur la pin 6
pour generer le signal PWM
}
delay (00) ; // delai de 100ms avant la nouvelle acquisition
}
```

Matlab et Arduino

Programmation de la carte Arduino Uno comme une carte d'interface.

Sous simulink

La bibliothèque Instrument Control Toolbox offre les blocs qui permettent l'échange des données binaires.



Matlab et Arduino

Programmation de la carte Arduino Uno comme une carte d'interface.

Sous simulink

Ces blocs sont les suivants :

- Serial Configuration : Configuration des paramètres du port série.
- Serial Send : Envoie des données binaires via le port série.
- Serial Receive : Acquisition des données binaires via le port série

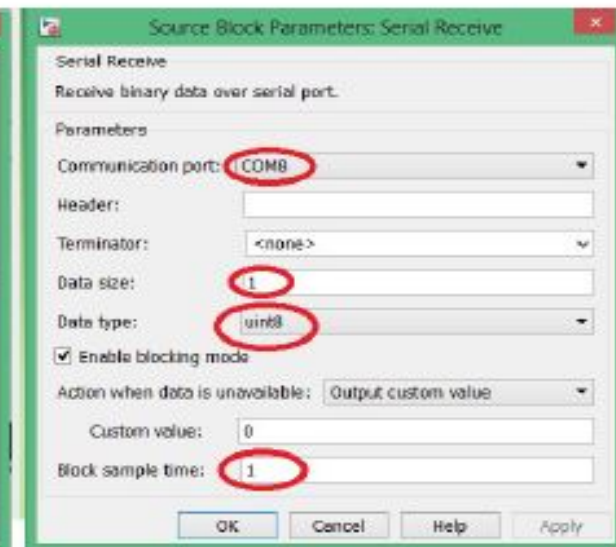
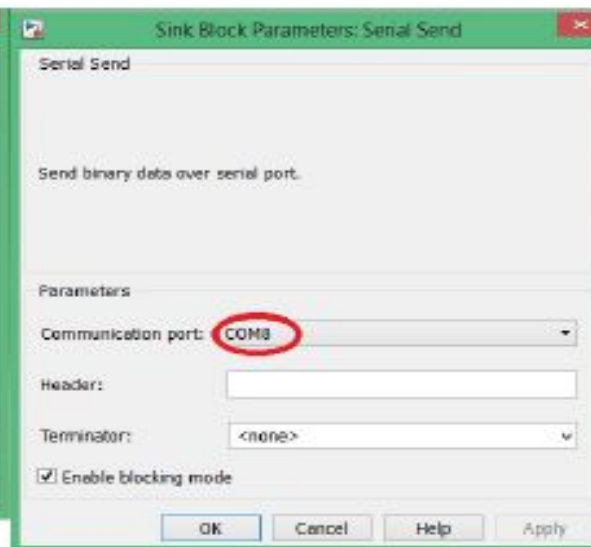
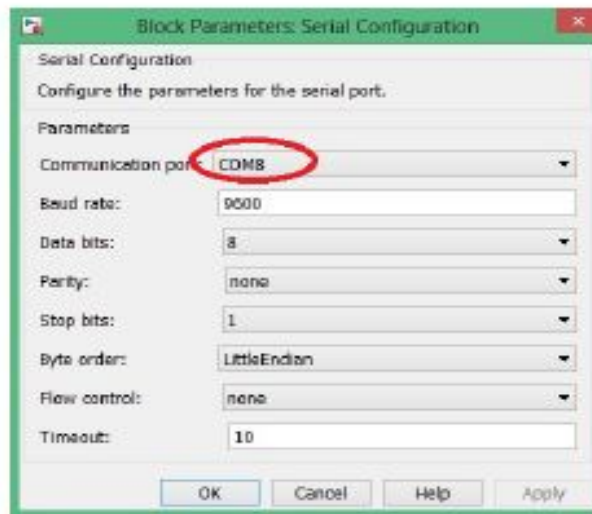
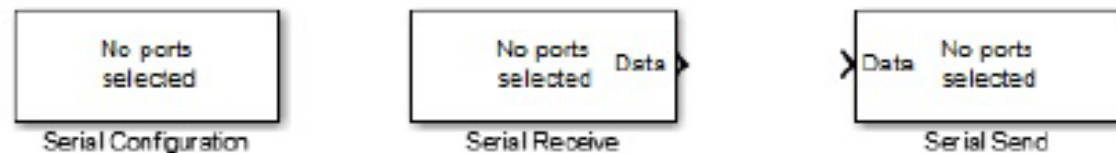
Matlab et Arduino

Programmation de la carte Arduino Uno comme une carte d'interface.

Sous simulink

Les paramètres à configurer sont :

- Communication Port
- Data size
- Data type
- Block sample time



Matlab et Arduino

Programmation de la carte Arduino Uno comme une carte d'interface.

Sous simulink

Exemple d'acquisition et d'envoi sous Simulink:



Matlab et Arduino

Utilisation du package ArduinoIO.

Cette solution consiste à utiliser la carte arduino comme une interface d'entrées(AnalogInput)/sorties(Analog/Digital Output). Ce package permet de communiquer Matlab ou Simulink avec la carte Arduino via un câble USB.

Elle consiste à pré-charger un programme dans la carte Arduino afin que celle-ci fonctionne en serveur.

Ce programme consiste à "écouter" les requêtes envoyées via la liaison série (USB) et de répondre à ces requêtes en renvoyant l'état d'une entrée ou en modifiant l'état d'une sortie. Ces mêmes entrées/sortie sont vues dans matlab comme des entrées logiques ou analogiques (utilisation du CAN) ou des sorties analogiques (mode PWM).

Matlab et Arduino

Utilisation du package ArduinoIO.

Pré-chargement du programme dans la carte Arduino

1. Télécharger le package ArduinoIO
2. Décompresser à la racine de votre disque dur, exemple E :\arduinoio
3. Ouvrir le dossier décompressé.
4. Aller vers : "ArduinoIO\pde\adiosrv" *
5. Charger le fichier adiosrv.pde vers le logiciel Arduino.
6. Televerser !

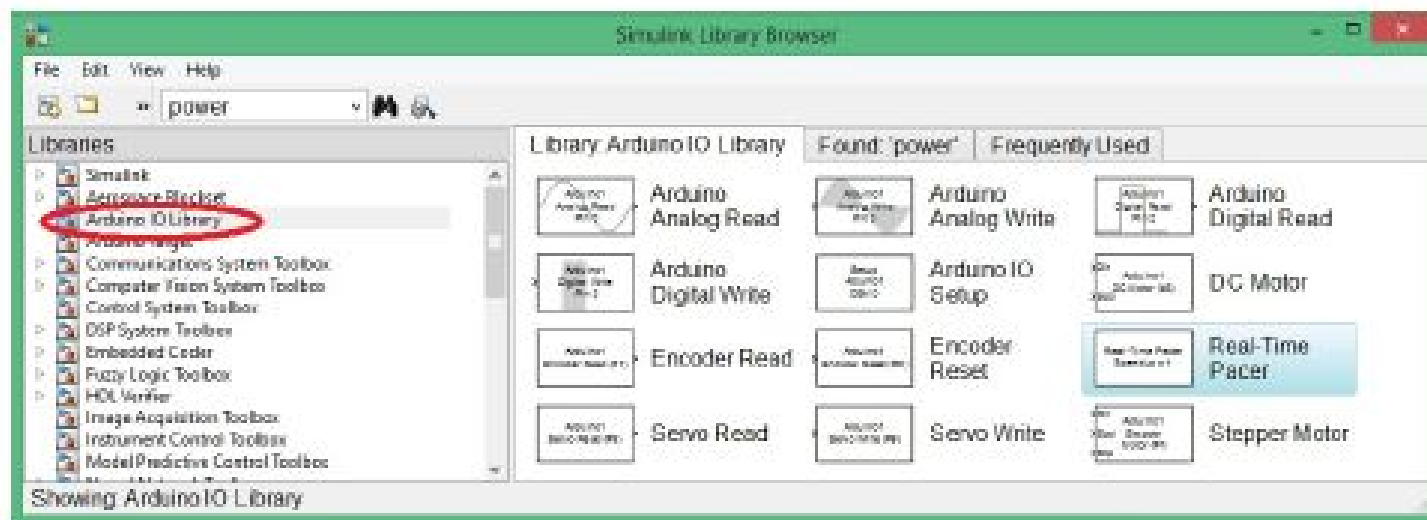
** adiosrv est l'abréviation de : Analog and Digital Input and Output Server for MATLAB. La carte Arduino UNO est maintenant configuré pour être utiliser comme une carte d'interface Entrées/Sorties.*

Matlab et Arduino

Utilisation du package ArduinoIO.

Installation du package ArduinoIO

1. Lancer Matlab et placer vous dans le répertoire E :\arduinoio
2. Exécuter la commande : install-arduino
3. Fermer et relancer Matlab puis Simulink
4. Dans les bibliothèques se trouvent maintenant les blocs dans Arduino IO library.



Matlab et Arduino

Utilisation du package ArduinoIO.

Exploitation de la bibliothèque ArduinoIO sous Simulink

Les blocs nécessaires pour notre objectif d'asservissement sont les suivants :

- **Real-Time Pacer:** Ce bloc permet de ralentir le temps de simulation de sorte qu'il synchronise avec le temps réel écoulé. Le coefficient de ralentissement est contrôlable par l'intermédiaire du paramètre Speedup
- **Arduino IO Setup:** Pour configurer sur quel port la carte Arduino UNO est connectée. Pour cela il suffit de voir dans Gestionnaire des périphériques. voir Figure 4.
- **Arduino Analog Read:** Pour configurer à partir de quel pin [0,1,2,3,4,5] on va acquérir les données du capteur.
- **Arduino Analog Write:** Pour configurer à partir de quel pin [3,5,6,9,10,11] on va envoyer la commande en PWM vers l'actionneur.



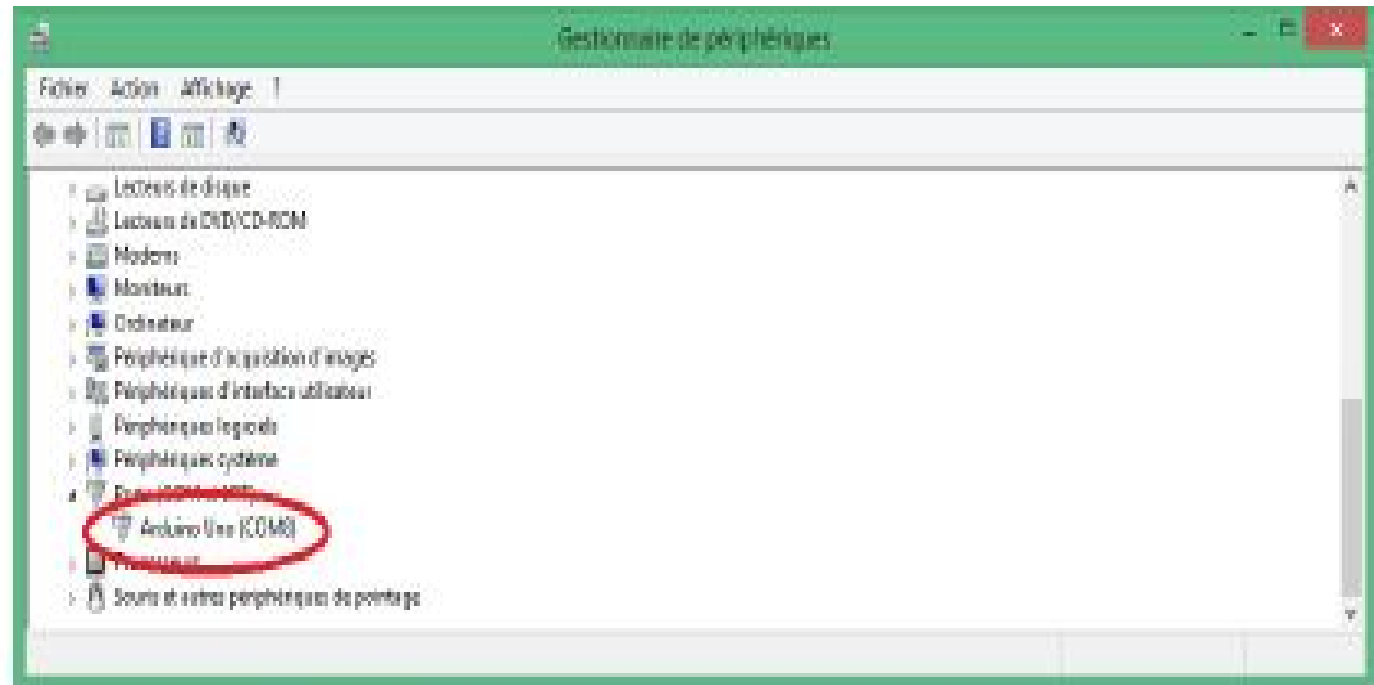
Matlab et Arduino

Utilisation du package ArduinoIO.

Exploitation du package ArduinoIO sous Matlab

Le package ArduinoIO offre une panoplie de commandes permettant d'écrire un programme sous Matlab (M-file). Pour accéder à ces commandes il faut créer un objet arduino dans l'espace de travail et spécifier le port sur lequel la carte arduino est connecté avec la commande :

>> a=arduino('port');



Matlab et Arduino

Utilisation du package ArduinoIO.

Exploitation du package ArduinoIO sous Matlab

Parmi les commandes qui sont accessibles on retrouve :

- **pinMode**

Exemple : `a.pinMode(11,'output')` // configurer la pin 11 comme sortie.

- **digitalRead**

Exemple : `val=a.digitalRead(4)` ; // lecture de l'état de la pin 4

- **digitalWrite**

Exemple : `a.digitalWrite(13,0)` ; // mettre la pin 13 à l'état bas 0V

- **analogRead**

Exemple : `val=a.analogRead(0)` ; // lecture de la pin 0 de l'ADC

- **analogWrite**

Exemple : `a.analogWrite(3,10)` ; // envoyer sur la pin 10 un signal pwm de rapport cyclique 10/255

Matlab et Arduino

Utilisation du package Arduino Target.

Embedded Coder Support Package for Arduino permet de créer des applications Simulink qui vont fonctionner de façon autonome sur la carte Arduino. on dit que la carte Arduino est devenue une cible (Target) et elle peut fonctionner d'une façon autonome (sans avoir recours à Matlab/Simulink).

Dans la suite, on utilisera les blocs Simulink offert par le package ArduinoIO Library et la librairie Instrument Control Toolbox pour l'acquisition et l'envoi des données.

Matlab et Arduino

Exemple 1 : Capteur à Ultrasons HC-SR04

Programmation de la carte Arduino UNO

```
#define echoPin 7 // Echo Pin  
#define trigPin 8 // Trigger Pin  
long duree , distance ;  
void setup ()  
{  
  Serial.begin (9600) ;  
  pinMode ( trigPin , OUTPUT) ;  
  pinMode ( echoPin , INPUT) ;  
}
```

Matlab et Arduino

Exemple 1 : Capteur à Ultrasons HC-SR04

Programmation de la carte Arduino UNO (Suite)

```
void loop ()
{
//Envoyer l e s i g n a l sur la pin 8
digitalWrite ( trigPin , LOW) ;
delayMicroseconds (2) ;
digitalWrite ( trigPin , HIGH) ;
delayMicroseconds (10) ;
digitalWrite ( trigPin , LOW) ;
// avoir la duree en ms
duree = pulseIn ( echoPin , HIGH) ;
// Vitesse du Son 340 m/ s
// Calculer la distance ( en cm)
distance = ( duree /2) *340 *0.0001 ;
// envoie de la donnee sur l e port s e r i e
S e r i a l . write ( distance ) ;
// Delai de 50 ms avant la nouvelle a c q u i s i t i o n
delay (50) ;
}
```

Matlab et Arduino

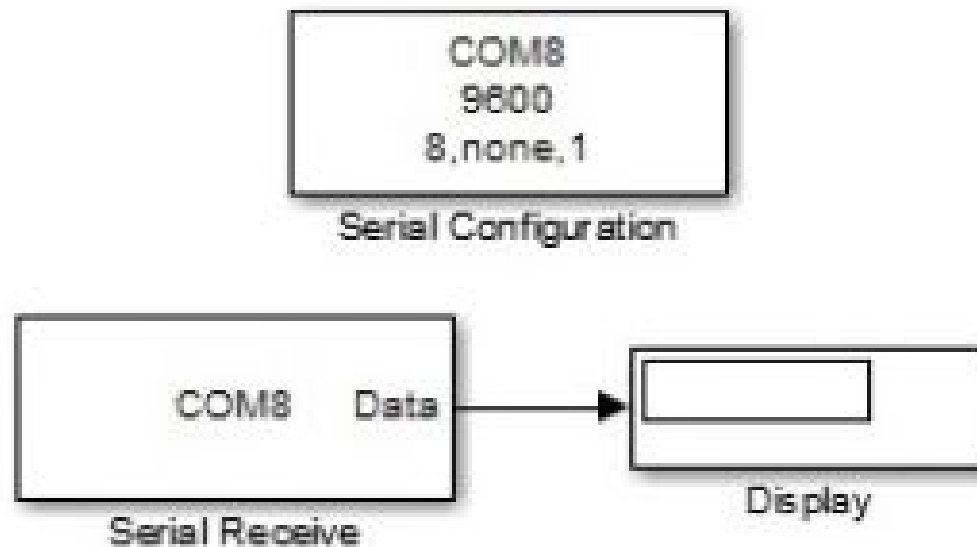
Exemple 1 : Capteur à Ultrasons HC-SR04

Développement du modèle Simulink :

Il suffit d'utiliser les blocs offerts par Instrument Control Toolbox.

- Un bloc Serial Configuration
- Un bloc Serial Receive
- Un bloc Display ou Scope

pour assurer la lecture de la distance en temps réel.



Matlab et Arduino

Exemple 2 : Capteur de température LM35

Programmation de la carte Arduino UNO

```
int temp ;
void setup ()
{
  Serial.begin (9600) ;
}
void loop ()
{
  // lecture de la donnée a partir du CAN ( valeur entre 0 et 1023)
  temp = analogRead (A0) ;
  // envoie de la donnée via le port serie
  Serial.write (temp) ;
  // delai de 1 s avant nouvelle acquisition
  delay (1000) ;
}
```

Matlab et Arduino

Exemple 2 : Capteur de température LM35

Développement du modèle Simulink :

Le modèle Simulink qui va traiter les données transmises à partir de la carte Arduino UNO.

